NEURAL DECODING WITH REDUNDANT AND RELIABILITY-CONDITIONED DUAL RELATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural decoders for linear codes typically treat the supplied parity-check matrix as a fixed structural description, even though a code admits many algebraically equivalent full-rank bases of the same dual space. We show that this choice is not neutral for finite neural decoders: equivalent parity-check bases can change bit-error rate (BER) by more than $7\times$ under matched code, rank, row space, architecture, and parameter count, with a residual gap remaining even under an exact row-weight-matched control. We then move beyond a single basis by exposing a rank-complete vocabulary of redundant dual relations and conditioning each relation on a normalized log soft-parity confidence derived from the current observation. Across Gallager LDPC, BCH, and MacKay LDPC, moving beyond repeated native parity coordinates to redundant dual-relation representations consistently improves a fixed CrossMPT backbone, while the benefit of particular relation sets is code dependent. Relation-level reliability provides an additional code-dependent gain and remains effective in a larger-blocklength replication. On MacKay LDPC, reassigning the same per-frame reliability values to incorrect equal-weight relation identities increases BER by roughly 51–94% across 3–5 dB, and paired rescue–harm analysis shows that reliability conditioning more often repairs than introduces frame errors. Together, these results show that neural decoding depends not only on which parity relations are exposed, but also on how their observation-dependent reliability is represented to the decoder.

1 INTRODUCTION

Neural channel decoders increasingly exploit explicit code structure rather than learning an unconstrained mapping from channel observations to codewords. Neural belief propagation (BP), graph/syndrome decoders, and Transformer-style decoders inject parity structure through Tanner-graph messages, masks, syndromes, or variable/check interactions (Nachmani et al., 2016; 2018; Bennatan et al., 2018; Cammerer et al., 2022; Choukroun & Wolf, 2022; Park et al., 2025b; 2026). Across these approaches, however, the particular parity representation given to the decoder is usually treated as fixed.

Consider a binary linear code $\mathcal{C}$ with block length n and k information bits, so $r = n - k$ independent parity constraints are required. A full-rank matrix $H \in \mathbb{F}_2^{r\times n}$, with $\mathbb{F}_2 = \{0, 1\}$, stores one independent set of these constraints as its rows. Their binary combinations form all valid parity relations, called the dual code $\mathcal{C}^\perp$. Hence the rows of H form one basis of $\mathcal{C}^\perp$, but not the only one. For any invertible binary $r \times r$ matrix A,

$$H' = AH \pmod 2, \qquad \mathcal{C}^\perp = \mathrm{rowspan}(H) = \mathrm{rowspan}(H').$$

Thus H and H' are algebraically equivalent descriptions of the same constraints. A neural decoder, however, receives the concrete rows, graph neighborhoods, masks, or relation tokens rather than the abstract row space. Algebraically equivalent bases can therefore become different neural representations.

This leads to our central question: *What parity representation should a neural decoder see?* In matched experiments, changing only the parity basis changes the bit-error rate (BER) by more than $7\times$, and a residual gap remains across a second neural backbone, a longer code, and an exact row-weight-matched control. Thus parity representation is itself a meaningful design variable.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Rather than searching for a single universally good basis, we move beyond one basis and represent the code using a rank-complete set of valid dual relations. The resulting redundant relation vocabulary preserves the same dual constraint space while exposing multiple coordinates through which the decoder can access it. Our experiments show that this richer representation can improve decoding, while the usefulness of individual relations remains code dependent.

A static relation set is still incomplete, however, because the reliability of a relation depends on the current noisy observation. For each selected dual relation u, a binary vector indicating the participating bits, we therefore aggregate the soft evidence in the received vector $\mathbf{y}$ over its support to obtain an observation-dependent reliability $\rho_u(\mathbf{y})$. This makes the representation sample dependent and separates the problem into two questions: *which relations should be exposed?* and *which of those relations should be trusted for the current observation?*

Across multiple code families, neural backbones, and a longer-blocklength setting, we find consistent evidence for this progression from basis choice to redundant relations and then to relation-conditioned reliability. Controlled interventions further show that the reliability signal is useful when it remains associated with the corresponding dual relation, rather than acting only as a global collection of confidence values.

Contributions. Our contributions are:

- **C1: Parity-representation sensitivity.** We show that algebraically equivalent parity-check bases can produce materially different neural decoding behavior, including under matched architecture, training, and parity-row controls.
- **C2: Redundant dual-relation representations.** We move beyond a single parity basis to a rank-complete vocabulary of redundant dual relations and show that this representation consistently improves over repeated native parity coordinates, while relation quality remains code dependent.
- **C3: Observation-conditioned relation reliability.** We introduce a sample-dependent reliability signal for each selected dual relation, enabling the decoder to represent both which relation is present and how strongly the current observation supports it.
- **Mechanism and robustness evidence.** Controlled ablations and relation–reliability interventions show that the gain depends on the correspondence between relation identity and observation-dependent reliability, and the overall effect persists across different code families, neural backbones, and a larger block length.

2 Related Work

Our work connects three lines of research: structure-aware neural decoding, parity-representation design, and reliability-aware decoding. Structure-aware neural decoders make the supplied parity representation part of the computation. Neural BP learns parameters on Tanner-graph messages (Nachmani et al., 2016; 2018), graph and syndrome decoders combine parity structure with soft evidence (Bennatan et al., 2018; Satorras & Welling, 2021; Cammerer et al., 2022), and Transformer decoders encode parity structure through masks or variable/check interactions (Choukroun & Wolf, 2022; Park et al., 2025b; 2026). Building on these directions, we study the parity representation itself as a design variable while holding the neural backbone and training protocol fixed.

Alternative and redundant parity representations have been effective across classical and neural decoding. Multiple-bases BP and automorphism ensembles exploit alternative parity graphs for the same code (Hehn et al., 2010; Geiselhart et al., 2022); redundant-check and dual-code methods use overcomplete or selected dual relations (Santi et al., 2018; Bossert, 2020); and neural BP can learn importance over overcomplete checks (Buchberger et al., 2021). Multiple-mask variants of the Error Correction Code Transformer (ECCT) and cyclically equivariant decoders likewise expose multiple parity-derived views (Park et al., 2025a; Chen & Ye, 2021). These works motivate a broader view of parity structure as a decoding design choice. We develop this view in a neural representation setting by explicitly separating equivalent-basis sensitivity, redundant-relation exposure, and relation-selection quality under matched neural capacity.

Reliability-aware decoding includes methods that adapt parity processing according to received soft information (Jiang & Narayanan, 2006; Santi et al., 2018), while CrossMPT provides bit-level magnitudes to a structure-aware neural decoder (Park et al., 2025b). Our formulation keeps the

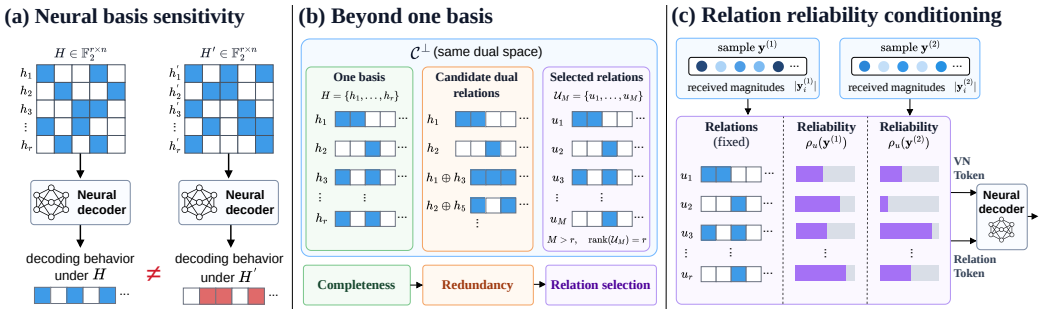


Figure 1: **From parity-basis sensitivity to reliability-conditioned dual relations.** (a) Algebraically equivalent parity bases can induce different neural representations. (b) We move beyond one basis by selecting additional valid dual relations while preserving the full dual space. (c) The selected relations are fixed, while observation-dependent reliability conditions each relation token on the current received sample.

relation vocabulary fixed and conditions each relation token on an observation-dependent reliability summary. This makes relation identity and current evidence explicit as separate pieces of information, leading naturally from *which relations are exposed* to *which exposed relations are trustworthy now*.

3 Problem Formulation

3.1 Linear Codes and Parity Representations

Using the notation above, let $\mathcal{C} \subseteq \mathbb{F}_2^n$ have dimension k and redundancy $r = n - k$. A full-rank parity-check matrix $H \in \mathbb{F}_2^{r \times n}$ satisfies

$$\mathcal{C} = \{\mathbf{x} \in \mathbb{F}_2^n : H\mathbf{x}^\top = \mathbf{0}\}, \qquad \mathcal{C}^\perp = \mathrm{rowspan}(H). \tag{1}$$

The rows of H are therefore one basis of the dual code, not the dual code itself. For any invertible binary matrix $A \in \mathbb{F}_2^{r \times r}$,

$$H' = AH \pmod 2 \tag{2}$$

spans the same $\mathcal{C}^\perp$ and defines exactly the same code. We call the set of parity relations exposed to a decoder its *parity representation* $\mathcal{R}$.

3.2 Channel Model and Representation-Dependent Decoding

For a codeword $\mathbf{x} \in \{0,1\}^n$, binary phase-shift keying (BPSK) maps each bit to $s_i = 1 - 2x_i$. Under an additive white Gaussian noise (AWGN) channel,

$$\mathbf{y} = \mathbf{s} + \mathbf{z}, \qquad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I), \tag{3}$$

where I is the $n \times n$ identity matrix and σ^2 is the noise variance.

We write a structure-aware neural decoder as

$$\hat{\mathbf{x}} = f_\theta(\mathbf{y}; \mathcal{R}), \tag{4}$$

to make explicit that decoding depends not only on the received observation $\mathbf{y}$, but also on the concrete parity representation supplied to the model. In existing structure-aware neural decoders, $\mathcal{R}$ is typically instantiated from a fixed parity-check matrix and is not itself treated as a design variable.

Our central problem is therefore to determine how the dual constraints of a fixed code should be represented to a neural decoder. We report channel quality using the signal-to-noise ratio (SNR) E_b/N_0 in decibels (dB).

3.3 Representation Ambiguity

Let $\mathcal{R}_H = \{h_1, \ldots, h_r\}$ denote the row basis of H and $\mathcal{R}_{H'} = \{h'_1, \ldots, h'_r\}$ the row basis of an equivalent H'. Coding theory regards these as equivalent descriptions of the same dual subspace.

A learned model, however, receives concrete rows, graph neighborhoods, masks, or tokens. It is therefore not guaranteed to be invariant to an arbitrary change of basis. Figure 1(a) visualizes this mismatch: the code is unchanged, but the concrete parity coordinates presented to the neural decoder differ. Section 6 measures this sensitivity under matched controls, motivating a redundant representation that still spans the same dual space.

4 Redundant Dual-Relation Decoding

Figure 1 gives an overview of the proposed progression: panel (a) motivates the representation problem, panel (b) introduces the redundant dual-relation vocabulary, and panel (c) adds sample-dependent relation reliability. Together, these panels move from *which basis?* to *which relations?* to *which relations should be trusted now?*

4.1 From a Basis to a Dual-Relation Vocabulary

As illustrated in Figure 1(b), we retain the full dual space while expanding the representation beyond one basis. Let M be the number of relation tokens and collect their parity rows as

$$U = [u_1^\top, \dots, u_M^\top]^\top \in \mathbb{F}_2^{M\times n}, \qquad u_m \in \mathcal{C}^\perp, \qquad \operatorname{rank}_{\mathbb{F}_2}(U) = r. \tag{5}$$

Thus U still spans the full dual space. For $M = r$ with independent rows it is a basis representation; for $M > r$ it can expose additional valid relations. We treat U as an ordered decoder input: rows may repeat in *NativeRepeat*, whereas the redundant conditions use distinct relations.

4.2 Candidate Generation and Relation Selection

Enumerating all 2^r dual codewords is unnecessary. We instead build finite candidate pools from low-order binary combinations of valid parity rows, remove duplicates, sort non-native candidates by increasing Hamming weight with a fixed deterministic tie-break, append rows until budget M is reached, and verify dual rank r and full bit coverage. Selection is completed before training and never uses BER, validation loss, or trained-model outputs. The exact combination orders and tie-breaks are benchmark specific and listed in Appendix A.1.

This rule is only one reproducible way to choose redundant relations, not a claim of universal optimality. To separate the effect of redundancy from the quality of a particular selector, the experiments also include controls whose row-weight distributions are matched while their non-native relation identities are changed. Benchmark-specific candidate pools, tie-breaks, and control constructions are specified in Section 5.

4.3 Observation-Conditioned Relation Reliability

Figure 1(c) depicts the second step: the selected relation identities remain fixed, but their relation-specific reliabilities vary with the received sample. We therefore aggregate the received magnitudes over the support of each relation.

For bit i, define

$$r_i = \tanh(\beta|y_i|),$$

where $\beta > 0$ is learned through a softplus parameterization. For relation u, let $\operatorname{supp}(u)$ denote the set of bit indices that participate in the parity relation, and let $|u| = |\operatorname{supp}(u)|$ be its Hamming weight. We define

$$\rho_u(\mathbf{y}) = \frac{1}{|u|} \sum_{i\in\operatorname{supp}(u)} \log r_i. \tag{6}$$

Because $0 < r_i < 1$, larger $\rho_u(\mathbf{y})$ (closer to zero) means stronger aggregate confidence in the participating bit evidence; whether the parity equation is satisfied is represented separately by the hard syndrome sign below. Normalization by $|u|$ reduces direct dependence on relation length.

The statistic also has a direct soft-decoding interpretation. Let $L_i = \log \frac{P(x_i=0|y_i)}{P(x_i=1|y_i)}$ denote a bit log-likelihood ratio (LLR). Then $\mathbb{E}[(-1)^{x_i} \mid y_i] = \tanh(L_i/2)$. Under the standard

independent-bit belief approximation, the magnitude of the soft confidence in parity relation u is $\prod_{i\in\mathrm{supp}(u)} \tanh(|L_i|/2)$; taking logs yields a sum of $\log\tanh(|L_i|/2)$. For BPSK over AWGN, $|L_i|$ is proportional to $|y_i|$, so the learned scale β absorbs the channel scaling and ρ_u can be read as a length-normalized log soft-parity confidence rather than an arbitrary auxiliary feature.

Importantly, u is fixed by the code representation, whereas $\rho_u(\mathbf{y})$ changes from sample to sample. Thus relation selection determines *which constraints are exposed*, while $\rho_u(\mathbf{y})$ describes *how trustworthy each exposed constraint is for the current observation*.

4.4 Reliability-Conditioned CrossMPT

Given a support matrix $U \in \{0,1\}^{M\times n}$, where $U_{u,i} = 1$ means that bit i participates in relation u, CrossMPT represents bits as variable-node (VN) tokens and parity relations as check-node (CN) tokens, then alternates information exchange between the two under the support mask (Park et al., 2025b). The support matrix therefore determines which bit and relation tokens may interact.

The initial bit token retains the standard magnitude input,

$$\mathbf{v}_i^{(0)} = |y_i|\mathbf{e}_i^{\mathrm{VN}}.$$

Let $\hat{x}_i^{\mathrm{hard}} = \mathbf{1}[y_i < 0]$ be the hard BPSK decision and define

$$s_u = 1 - 2\Big(\sum_{i\in\mathrm{supp}(u)} \hat{x}_i^{\mathrm{hard}} \bmod 2\Big) \in \{-1,+1\}.$$

Thus s_u records hard parity satisfaction/violation, whereas $\rho_u(\mathbf{y})$ records confidence in the participating bit evidence. The baseline relation token is

$$\mathbf{c}_u^{(0)} = s_u\mathbf{e}_u^{\mathrm{CN}}.$$

Here $\mathbf{e}_i^{\mathrm{VN}}$ and $\mathbf{e}_u^{\mathrm{CN}}$ are learned token embeddings.

We condition this token on the current observation by adding the relation reliability:

$$\mathbf{c}_u^{(0)} \leftarrow \mathbf{c}_u^{(0)} + W_\rho\,\rho_u(\mathbf{y}), \tag{7}$$

where W_ρ is a bias-free scalar-to-hidden projection.

The backbone and attention topology are otherwise unchanged. Consequently, the comparison isolates the effect of the parity representation and relation-level conditioning rather than introducing a new message-passing architecture.

4.5 Implementation

All controlled comparisons use the same CrossMPT backbone and training protocol within each benchmark. Adding ρ requires only a scalar-to-hidden projection and one learned scalar β; the detailed hyperparameters, checkpoint schedule, parameter counts, and multiply–accumulate (MAC) estimates are given in Appendix A.2.

5 Experimental Setup

5.1 Benchmarks and Representation Conditions

We denote a code by (n,k), with block length n and k information bits. Learned$(31,16)$ is the fixed short-code matrix used for controlled ablations. The main benchmarks are Gallager LDPC$(96,50)$, BCH$(63,36)$, and a fixed MacKay LDPC$(96,48)$ alist instance; all underlying matrices remain fixed across representation conditions. Gallager LDPC$(252,128)$ is used only as a larger-blocklength replication. Matrix provenance and construction details are summarized in Appendix A.1.

The main long-code studies use a common relation budget $M = 128$. We define three principal representation conditions. ***NativeRepeat*** repeats the rows of the native parity-check representation until all M relation slots are filled; the native row list may itself contain linear dependencies, as

in the Gallager constructions. *Selected Redundant* replaces repeated slots with distinct valid dual relations chosen by the deterministic low-weight procedure in Section 4. *Selected Redundant+ρ* uses exactly the same selected relations and additionally conditions each relation token on its observation-dependent reliability. The Gallager$(252, 128)$ scale study uses $M = 256$ and the same three-way ladder.

The Selected supports are generated before training from valid parity relations only; exact candidate pools and deterministic tie-breaks are listed in Appendix A.1. No selector is tuned using BER or trained-model outputs.

We also use matched controls to distinguish redundancy from relation choice. *Random-WM* means *random weight-matched*: it keeps the same relation budget, full rank, native rows, uniqueness, candidate family, and exact row-weight histogram as Selected, while changing which non-native relations are used. A second inference-only control, SAMEWEIGHTSHUFFLE, keeps a trained Redundant+ρ model fixed but permutes reliability values only among relations with the same Hamming weight. This preserves the per-frame reliability distribution within each weight class while breaking the exact relation–reliability correspondence.

5.2 TRAINING, BACKBONE REFERENCE, AND METRICS

CrossMPT is the backbone for the main C2/C3 comparisons; ECCT is an exact reproduced second-backbone C1 check, not an ECCT-style surrogate. BER is the fraction of decoded bits in error, and frame-error rate (FER) is the fraction of decoded codewords containing at least one bit error. Learned$(31, 16)$ controls use seeds 42–44 and 30k steps. Gallager and BCH use three seeds and 500k checkpoints; MacKay uses seeds 42–44 and a 300k validation-selected protocol. The Gallager$(252, 128)$ scale replication uses seeds 101–103 and 300k steps. Final paired evaluations use common message/noise streams, with 131,072 random codewords per seed and E_b/N_0 point for the long-code comparisons (163,840 for the short C1 replications). Training spans 2–7 dB. Full checkpoint, validation, batch-size, and efficiency details are in Appendix A.2. A "3–5 dB mean" is the arithmetic BER mean over $E_b/N_0 \in \{3, 4, 5\}$ dB within each seed. Percentage gains are seed-wise relative BER reductions averaged across seeds, so they need not equal ratios of the rounded seed-averaged entries in Table 1.

5.3 PAIRED ERROR AND RELIABILITY-ALIGNMENT EVALUATION

For this analysis, *Redundant-None* denotes the Selected Redundant model without relation reliability. Redundant-None and Redundant+ρ are evaluated on identical message/noise streams, and frames are partitioned into *easy* (both correct), *rescue* (None fails, ρ succeeds), *harm* (None succeeds, ρ fails), and *hard* (both fail) groups. This paired evaluation identifies where relation-level conditioning helps, but the group statistics are treated as associations rather than causal evidence.

To test whether the gain depends on relation-specific reliability rather than merely a frame-level confidence signal, we additionally intervene on a trained Redundant+ρ model at inference time. Let $\boldsymbol{\rho}_b = (\rho_{b,1}, \ldots, \rho_{b,M})$ denote the reliability vector for frame b. The primary control compares the original assignment

$$\text{TRUE}: \quad \rho'_{b,j} = \rho_{b,j} \tag{8}$$

with a within-frame permutation

$$\text{WITHINSHUFFLE}: \quad \rho'_{b,j} = \rho_{b,\pi_b(j)}. \tag{9}$$

The permutation preserves the complete multiset of reliability values in each frame, including its mean and quantile spread, while destroying the relation–reliability correspondence. Our stricter SAMEWEIGHTSHUFFLE control restricts the permutation to indices j, k satisfying $|u_j| = |u_k|$. For MacKay, the 128 selected relations form weight-6 and weight-10 groups of sizes 48 and 80; for Gallager, they form weight-6, weight-8, and weight-10 groups of sizes 49, 32, and 47. Thus every relation is shufflable on both codes. We use a nonzero cyclic permutation within each group for every frame, preserving the exact per-frame reliability multiset inside each weight class with no cross-weight moves while breaking exact relation identity.

Table 1: **Cross-code representation benchmark at representative E_b/N_0 points.** BER is lower-is-better and entries are means over the three measured training seeds. Each long-code seed and E_b/N_0 point uses 131,072 evaluation frames. **Bold** and underlining denote the lowest and second-lowest BER, respectively, among the measured neural representation variants within each code block and column. MacKay Random-WM exactly matches Selected Redundant in relation budget, rank, uniqueness, native rows, candidate family, and row-weight histogram; the stricter paired Gallager Random-WM control is reported in Sec. 6.2. Percentage gains quoted in the text are seed-wise relative reductions averaged over seeds, not ratios of these rounded mean-BER entries.

Code	Decoder / representation	M	3 dB	4 dB	5 dB	3–5 mean
Gallager LDPC $(96, 50)$	CrossMPT Native	48	1.54×10^{-2}	3.59×10^{-3}	5.46×10^{-4}	6.51×10^{-3}
	NativeRepeat	128	1.43×10^{-2}	3.09×10^{-3}	4.36×10^{-4}	5.94×10^{-3}
	Selected Redundant	128	1.12×10^{-2}	2.11×10^{-3}	2.85×10^{-4}	4.54×10^{-3}
	Ours: Selected Redundant+ρ	128	$\mathbf{8.31\times10^{-3}}$	$\mathbf{1.35\times10^{-3}}$	$\mathbf{1.89\times10^{-4}}$	$\mathbf{3.28\times10^{-3}}$
BCH $(63, 36)$	CrossMPT Native	27	3.81×10^{-2}	1.53×10^{-2}	3.96×10^{-3}	1.91×10^{-2}
	NativeRepeat	128	3.54×10^{-2}	1.32×10^{-2}	3.16×10^{-3}	1.73×10^{-2}
	Selected Redundant	128	2.07×10^{-2}	4.61×10^{-3}	4.94×10^{-4}	8.61×10^{-3}
	Ours: Selected Redundant+ρ	128	$\mathbf{1.64\times10^{-2}}$	$\mathbf{3.07\times10^{-3}}$	$\mathbf{2.58\times10^{-4}}$	$\mathbf{6.57\times10^{-3}}$
MacKay LDPC $(96, 48)$	CrossMPT Native	48	6.46×10^{-3}	6.40×10^{-4}	2.18×10^{-5}	2.37×10^{-3}
	NativeRepeat	128	5.994×10^{-3}	5.508×10^{-4}	1.743×10^{-5}	2.187×10^{-3}
	Random-WM	128	$\mathbf{4.896\times10^{-3}}$	$\mathbf{3.716\times10^{-4}}$	1.168×10^{-5}	$\mathbf{1.760\times10^{-3}}$
	Selected Redundant	128	5.434×10^{-3}	4.565×10^{-4}	1.391×10^{-5}	1.968×10^{-3}
	Ours: Selected Redundant+ρ	128	5.082×10^{-3}	4.139×10^{-4}	$\mathbf{9.855\times10^{-6}}$	1.835×10^{-3}

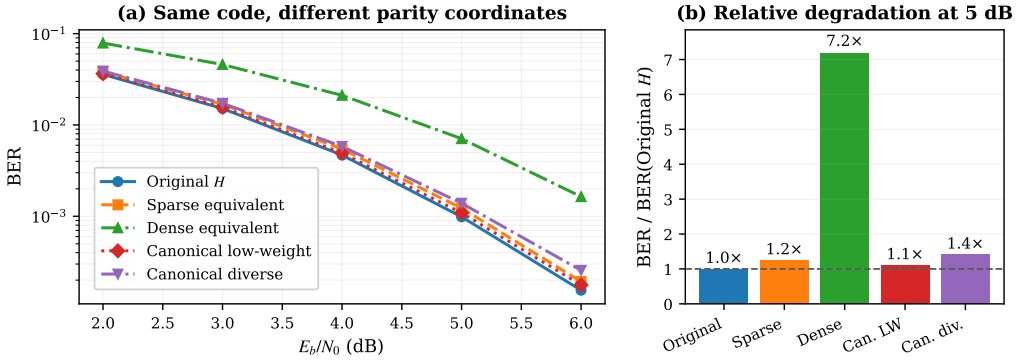


Figure 2: **Equivalent-basis sensitivity.** Five full-rank representations span the same dual space and use the same CrossMPT architecture and parameter count, yet yield materially different BER.

6 Results

We organize the evidence around basis sensitivity (C1), redundant relation representation (C2), relation-level reliability (C3), and its alignment mechanism. Table 1 is the primary cross-code benchmark; secondary diagnostics are deferred to the appendix.

6.1 Parity Representation Matters

Figure 2 compares five full-rank parity representations of Learned$(31, 16)$ while holding the code, dual rank, row space, bit coverage, CrossMPT architecture, parameter count, seeds, and training budget fixed. Despite this algebraic equivalence, the BER curves differ substantially: at 5 dB, the dense equivalent basis is $7.2\times$ worse than the original H, while the canonical low-weight and diverse spanning bases also fail to consistently improve upon the native representation.

This establishes concrete parity coordinates as a non-neutral neural interface. The same original/dense pair also degrades ECCT by $3.62 \pm 0.03\times$ at 5 dB across seeds 42–44, arguing against a CrossMPT-specific artifact (Appendix Table 6).

A paired Gallager LDPC$(96, 50)$ replication gives a $6.87\times$ 3–5 dB mean degradation for a dense equivalent basis; matched controls and the row-density caveat are detailed in Appendix A.8. More

stringently, on Learned$(31, 16)$ we construct a second full-rank basis with the same $M = 15$, rank, row space, bit coverage, and exact row-weight multiset $\{3{:}3, 5{:}7, 6{:}2, 7{:}3\}$. It is still $1.21\times$ worse over the 3–5 dB mean, with all three measured training seeds showing the same direction (Appendix Table 12 and Appendix A.9). This rules out row-weight distribution as a complete explanation of C1; remaining differences in column degree and relation intersections are documented in Appendix A.9.

6.2 Beyond a Single Basis: Redundancy and Relation Selection

The Learned$(31, 16)$ M-transition diagnostic also shows that selector preference changes with representation budget: relative to the low-weight selector, the diversity-oriented selector is 10.73% worse at $M = 15$, 37.43% better at $M = 32$, and 13.57% better at $M = 128$ (Appendix Table 3). This rules out a universal "diversity is better" heuristic before we turn to the longer-code controls below.

On Gallager, the stricter paired common-stream re-evaluation directly compares NativeRepeat, Random-WM, and Selected on the same messages and AWGN samples. Over 3–5 dB, Random-WM improves over NativeRepeat by 23.25%, while Selected improves by 23.80%; their paired mean differs by only 0.71%. Seed-wise standard deviations are reported in the appendix rather than emphasized in the headline comparison. Both matched redundant supports have the exact $\{6{:}49, 8{:}32, 10{:}47\}$ row-weight histogram, $M = 128$, rank 46, and the same dual row space. Thus the Gallager gain is primarily a redundancy effect rather than evidence for a uniquely strong selector (Appendix Table 5).

BCH Selected Redundant improves over NativeRepeat by 49.58%, while MacKay Selected improves by 9.97% (Appendix Table 4); detailed MacKay seed-wise variability is reported in Appendix A.10. On MacKay, Random-WM matches Selected in $M = 128$, rank, uniqueness, native rows, candidate family, and the exact 48 weight-6/80 weight-10 histogram while randomizing the 80 non-native identities. It improves over NativeRepeat by $19.49\% \pm 2.96\%$ and outperforms Selected on all three training seeds; the Random-WM$\rightarrow$Selected relative reduction is $-11.85\% \pm 2.31\%$ (Appendix A.10 and Table 4). At fixed training seed 42, five independently drawn matched Random-WM supports all beat Selected (mean 3–5 dB BER 1.771×10^{-3} vs. 2.010×10^{-3}), excluding a single lucky support draw (Appendix A.10). Together, Gallager and MacKay show that redundancy beyond repeated coordinates is robustly useful, whereas the value of a particular relation-selection rule is code dependent.

The larger-blocklength Gallager$(252, 128)$ replication gives the same qualitative C2 result at $M = 256$: over the 3–5 dB seed-level mean, replacing repeated native coordinates with 256 unique dual relations reduces BER by $21.22\% \pm 0.26\%$, with all three training seeds positive. This redundant support is not sparser than NativeRepeat (mean row weight 7.73 versus 6.00), so the scale result is not explained by choosing lighter checks. Full E_b/N_0-resolved values are in Appendix Table 13.

The Gallager and BCH gaps persist through the measured 500k-step horizon, while the MacKay models continue to improve beyond their earlier 100k checkpoints under the same-stream reevaluation. These optimization diagnostics are kept in Appendix Table 7 and Appendix A.10 rather than repeated in the main comparison.

6.3 Observation-Conditioned Relation Reliability

Table 1 shows a complementary improvement when relation-level reliability is added to the same selected support on all three main benchmarks. The 3–5 dB mean C3 reduction is 27.48% on Gallager, 23.29% on BCH, and 6.68% on MacKay; these seed-wise effect estimates are summarized in Appendix Table 4. The MacKay effect is more modest and more variable across seeds, but its seed-averaged BER improves at each reported E_b/N_0 point and all three seeds retain a positive 3–5 dB C3 gain. In the separate Gallager$(252, 128)$ scale replication, adding ρ to the same redundant support gives a further 33.80% 3–5 dB reduction and a 47.85% total reduction relative to NativeRepeat, again positive for all three seeds (Appendix Table 13). Because CrossMPT already receives bit magnitudes, the relevant question is not whether the decoder has access to channel reliability at all, but whether explicitly aggregating that evidence at the relation level is useful.

Table 2 provides the minimal capacity control. The capacity placebo adds the same 33 trainable parameters as the ρ pathway but receives no observation-dependent relation statistic. It leaves per-

formance unchanged, whereas CrossMPT+ρ improves 3–5 dB BER by 11.22% on Learned$(31, 16)$, ruling out added parameter capacity as the explanation. Alternative reliability aggregations are reported in Appendix Table 9; MeanLog is used for relation-length-normalized scale rather than claimed as uniquely optimal.

Table 2: **Compact C3 control on Learned**$(31, 16)$**.** The capacity placebo adds the same 33 trainable parameters as the ρ pathway but receives no observation-dependent relation statistic. CrossMPT+ρ keeps the original topology and adds the relation-indexed reliability feature.

Variant	3–5 dB BER	Gain	Params
CrossMPT	3.39×10^{-3}	–	31,585
Capacity placebo	3.39×10^{-3}	0.00%	31,618
CrossMPT+ρ	3.05×10^{-3}	11.22%	31,618

Adding ρ itself is computationally cheap relative to enlarging the relation vocabulary: on the Gallager benchmark, the redundant model uses 8.68 million analytic MACs per codeword and Redundant+ρ uses 8.70 million. Full efficiency, latency, and ECCT backbone-reference results are reported in Appendix Table 8.

6.4 Mechanism: Reliability Must Match the Relation

We evaluate mechanism evidence in two complementary ways. First, paired rescue/harm analysis shows that reliability conditioning repairs more baseline failures than it creates: over 3–5 dB the rescue-to-harm ratios are $8.37\times$, $21.25\times$, and $2.96\times$ on Gallager, BCH, and MacKay, with positive net FER rescue on all three codes (Appendix Table 10). This identifies where C3 helps, while the intervention below tests the relation–reliability correspondence directly.

The strongest mechanism test is the cross-code same-weight relation-alignment intervention. SameWeightShuffle keeps the trained checkpoint, received frames, and the exact per-frame reliability multiset within each equal-weight relation group fixed, while changing only which same-weight relation receives which reliability value. The correctly aligned True condition is better for every seed–E_b/N_0 pair: restoring alignment reduces BER by 56.6–96.2% on MacKay and 74.6–94.5% on Gallager over 3–5 dB. Detailed values are reported in Appendix Table 11. The result controls directly for relation Hamming weight and shows that the exact relation–reliability association matters, not merely the frame-level distribution of confidence values.

7 Conclusion

This work treats the parity representation supplied to a neural decoder as a design variable rather than fixed bookkeeping. Row-equivalent parity-check matrices define the same linear code, yet a finite learned decoder operates on concrete rows, supports, masks, and relation tokens. Our experiments show that equivalent bases can yield substantially different decoding behavior even when the code, dual space, architecture, parameter count, and training protocol are fixed. This motivates designing the representation of dual constraints rather than replacing one canonical basis with another.

We therefore move beyond a single basis and expose a rank-complete redundant collection of valid dual relations. The results separate redundancy from relation quality: additional coordinates can help, while the value of particular relations remains code dependent. We then attach to each selected relation an observation-dependent reliability derived from the current channel evidence. The mechanism analyses indicate that this gain comes from assigning confidence to the correct relation rather than from extra parameters or a global confidence scalar alone.

Taken together, the paper suggests a simple representation-level principle: expose useful dual relations and represent how trustworthy each relation is for the current observation. The larger-blocklength experiment preserves this progression, while scalable relation generation, adaptive selection, pruning under fixed computation, and stronger invariance to equivalent parity coordinates remain open. Our results establish this principle without claiming a universal selector or asymptotic scaling law.

REFERENCES

Amir Bennatan, Yoni Choukroun, and Pavel Kisilev. Deep learning for decoding of linear codes—a syndrome-based approach. In *IEEE International Symposium on Information Theory (ISIT)*, 2018. doi: 10.1109/ISIT.2018.8437530.

Martin Bossert. On decoding using codewords of the dual code, 2020. URL https://arxiv.org/abs/2001.02956.

Andreas Buchberger, Christian Hager, Henry D. Pfister, Laurent Schmalen, and Alexandre Graell i Amat. Pruning and quantizing neural belief propagation decoders. *IEEE Journal on Selected Areas in Communications*, 39(7):1957–1966, 2021. doi: 10.1109/JSAC.2020.3041392.

Sebastian Cammerer, Jakob Hoydis, Faycal Ait Aoudia, and Alexander Keller. Graph neural networks for channel decoding. In *IEEE Globecom Workshops (GC Wkshps)*, 2022. doi: 10.1109/GCWkshps56602.2022.10008601.

Xiangyu Chen and Min Ye. Cyclically equivariant neural decoders for cyclic codes. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1771–1780, 2021. URL https://proceedings.mlr.press/v139/chen21w.html.

Yoni Choukroun and Lior Wolf. Error correction code transformer. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38695–38705, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/fcd3909db30887ce1da519c4468db668-Abstract-Conference.html.

Marvin Geiselhart, Moustafa Ebada, Ahmed Elkelesh, Jannis Clausius, and Stephan ten Brink. Automorphism ensemble decoding of quasi-cyclic ldpc codes by breaking graph symmetries. *IEEE Communications Letters*, 26(8):1705–1709, 2022. doi: 10.1109/LCOMM.2022.3174164.

Thorsten Hehn, Johannes B. Huber, Olgica Milenkovic, and Stefan Laendner. Multiple-bases belief-propagation decoding of high-density cyclic codes. *IEEE Transactions on Communications*, 58(1):1–8, 2010. doi: 10.1109/TCOMM.2010.01.070468.

Jing Jiang and Krishna R. Narayanan. Iterative soft-input soft-output decoding of reed–solomon codes by adapting the parity-check matrix. *IEEE Transactions on Information Theory*, 52(8):3746–3756, 2006. doi: 10.1109/TIT.2006.878176.

Eliya Nachmani, Yair Be'ery, and David Burshtein. Learning to decode linear codes using deep learning. In *54th Annual Allerton Conference on Communication, Control, and Computing*, pp. 341–346, 2016. doi: 10.1109/ALLERTON.2016.7852251.

Eliya Nachmani, Elad Marciano, Loren Lugosch, Warren J. Gross, David Burshtein, and Yair Be'ery. Deep learning methods for improved decoding of linear codes. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):119–131, 2018. doi: 10.1109/JSTSP.2017.2788405.

Seong-Joon Park, Hee-Youl Kwak, Sang-Hyo Kim, Sunghwan Kim, Yongjune Kim, and Jong-Seon No. Multiple-masks error correction code transformer for short block codes. *IEEE Journal on Selected Areas in Communications*, 43(7):2518–2529, 2025a. doi: 10.1109/JSAC.2025.3559154.

Seong-Joon Park, Hee-Youl Kwak, Sang-Hyo Kim, Yongjune Kim, and Jong-Seon No. Crossmpt: Cross-attention message-passing transformer for error correcting codes. In *International Conference on Learning Representations*, 2025b. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/8f0942c43fcfba4cc66a859b9fcb1bba-Abstract-Conference.html.

Seong-Joon Park, Taewoo Park, Hee-Youl Kwak, Sang-Hyo Kim, Yongjune Kim, and Jong-Seon No. Efficient message-passing transformer for error correcting codes. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Xk8cwnwu2e.

Elia Santi, Christian Hager, and Henry D. Pfister. Decoding reed–muller codes using minimum-weight parity checks. In *IEEE International Symposium on Information Theory (ISIT)*, pp. 1296–1300, 2018. doi: 10.1109/ISIT.2018.8437637.

Victor Garcia Satorras and Max Welling. Neural enhanced belief propagation on factor graphs. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 685–693, 2021. URL https://proceedings.mlr.press/v130/garcia-satorras21a.html.

A Additional Experiments and Diagnostics

The main paper retains only experiments that directly establish the representation, reliability, or alignment claims. This appendix collects robustness checks, formulation choices, optimization details, and efficiency measurements that are not required to read the central argument.

Reproducibility/AI use. Full experimental protocols and result tables are provided in Section 5 and the appendix. The source code will be released on GitHub upon acceptance. Large language models were used for manuscript refinement and code debugging.

A.1 Selector and Representation Diagnostics

The benchmark matrices are fixed before any representation intervention: Learned$(31, 16)$ uses one fixed short-code parity matrix throughout, MacKay$(96, 48)$ uses one fixed alist instance throughout, and the Gallager/BCH constructions are deterministic. The ECCT C1 check exactly reproduces the referenced ECCT architecture and mask construction under the paired protocol. The deterministic low-weight selector applies low-order XOR candidate generation, duplicate removal, weight-first ordering, a fixed per-benchmark tie-break, and rank/coverage audit; supports are frozen before training and reused across seeds. Gallager$(96, 50)$ uses native rows plus unique pairwise XORs with lexicographic binary-row order as the equal-weight tie-break. MacKay$(96, 48)$ uses native rows plus unique pairwise XORs with a SHA-1 hash of the row bitstring as the deterministic tie-break. Gallager$(252, 128)$ retains its 126 native weight-6 checks and adds 130 pairwise-XOR relations, ordered by weight with the same SHA-1 tie-break. BCH$(63, 36)$ uses its fixed canonical binary basis and enumerates order-1, order-2, and order-3 XOR combinations before duplicate removal and weight-first selection. The basis and selector studies include transitions over $M \in \{15, 20, 32, 64, 128\}$ and comparisons among low-weight and diversity-oriented relation sets. Their qualitative conclusion is negative but useful: neither larger M, lower relation weight, nor greater diversity is universally optimal across codes and budgets. The exact row-weight-matched controls reinforce this point. On Gallager, paired common-stream evaluation finds only a $0.71\% \pm 0.40\%$ 3–5 dB mean gap between Selected and Random-WM, whereas on MacKay a matched random support outperforms the heuristic Selected support across training seeds. The fixed-seed five-support study below additionally shows that this MacKay advantage is stable across independently drawn matched supports. These experiments motivate treating relation selection as code dependent rather than elevating one heuristic to the method claim.

Table 3: **Representative points from the Learned$(31, 16)$ M-transition selector diagnostic.** Values are the BER reduction of the diversity-oriented selector relative to the low-weight selector; negative means diversity is worse. The non-monotonic sign and magnitude motivate treating selector preference as budget dependent rather than universal.

M	15	32	128
Diversity vs. low-weight	-10.73%	$+37.43\%$	$+13.57\%$

Table 4: **Seed-wise 3–5 dB representation effects used in the main text.** Positive values denote BER reduction. Gallager Random-WM/Selected uses the paired common-stream reevaluation; MacKay uses the 300k-horizon validation-selected protocol.

Code	Effect	3–5 dB BER reduction
Gallager$(96, 50)$	NativeRepeat→Selected	$23.80 \pm 0.15\%$
BCH$(63, 36)$	NativeRepeat→Selected	49.58%
MacKay$(96, 48)$	NativeRepeat→Selected	$9.97 \pm 3.07\%$
MacKay$(96, 48)$	NativeRepeat→Random-WM	$19.49 \pm 2.96\%$
Gallager$(96, 50)$	Selected→Selected+ρ	27.48%
BCH$(63, 36)$	Selected→Selected+ρ	23.29%
MacKay$(96, 48)$	Selected→Selected+ρ	$6.68 \pm 4.90\%$

Table 5: **Gallager paired selector control.** All rows use the same 500k checkpoints and a common evaluation stream. Selected and Random-WM have identical M, rank, row space, uniqueness, and row-weight histogram.

Comparison	3–5 dB BER reduction
NativeRepeat→Random-WM	$23.25 \pm 0.25\%$
NativeRepeat→Selected	$23.80 \pm 0.15\%$
Random-WM→Selected	$0.71 \pm 0.40\%$

Table 6: **Second-backbone equivalent-basis check on Learned$(31, 16)$ at 5 dB.** Both backbones use the exact same original H and dense row-equivalent H' pair. The ECCT row uses the reproduced ECCT architecture and mask construction under the same paired training protocol; values are means over seeds 42–44 and the degradation reports the mean seed-wise ratio. This control tests architecture generality and is intentionally kept separate from Table 1.

Backbone	Original H	Dense equivalent H'	Degradation
CrossMPT	9.87×10^{-4}	7.08×10^{-3}	$7.2\times$
ECCT	2.612×10^{-3}	9.452×10^{-3}	$3.62 \pm 0.03\times$

A.2 Long-Training Persistence, Hyperparameters, and Efficiency

Gallager and BCH are evaluated at 30k, 100k, 300k, and 500k checkpoints. Their combined improvement from NativeRepeat to Redundant+ρ persists across this measured horizon (Table 7). MacKay follows a separate 300k training and validation-selected evaluation protocol described in Appendix A.10.

Table 7: **Long-training persistence.** Combined NativeRepeat→Redundant+ρ BER reduction (%) over 3–5 dB. Mean $\pm$ std across three training seeds. The sparse four-checkpoint series is kept as a table rather than visually smoothed.

Code	30k	100k	300k	500k
Gallager	41.78 ± 0.92	42.97 ± 0.37	44.05 ± 0.58	44.64 ± 0.33
BCH	62.67 ± 0.47	60.52 ± 0.63	60.93 ± 0.69	61.33 ± 0.28

Gallager and BCH use hidden dimension 32 with two cross-attention layers in the long-training study. MacKay uses the validated larger CrossMPT configuration with hidden dimension 128, eight attention heads, and six layers; all MacKay representation conditions share this architecture. Gallager/BCH use Adam with learning rate 10^{-3} through 30k steps followed by cosine decay to 10^{-5} at 500k steps, while MacKay uses its separate 300k validation-selected protocol. Training E_b/N_0 is sampled uniformly from 2–7 dB. Adding ρ introduces one learned scalar β and a bias-free scalar-to-hidden projection; at hidden dimension 32 this is exactly 33 additional trainable parameters.

Table 8: **Protocol-matched Gallager backbone reference and efficiency profile at 30k steps.** BER/FER are 3–5 dB means over five fresh seeds. ECCT Native is included as a backbone reference; the representation interventions use CrossMPT. MACs are analytic per-codeword counts from the same implementation. Batch-1 latency and CUDA allocation are within-run measurements.

Method	M	Unique	Rank	3–5 BER	3–5 FER	Params	MACs	Latency	Peak mem.
ECCT Native	–	–	–	1.186×10^{-2}	0.36394	44,097	6.21M	1.618 ms	18.93 MiB
CrossMPT Native	48	48	46	7.225×10^{-3}	0.22326	44,130	4.74M	3.235 ms	17.91 MiB
CrossMPT Native+ρ	48	48	46	5.735×10^{-3}	0.17623	44,163	4.74M	3.307 ms	17.91 MiB
CrossMPT Redundant	128	128	46	5.199×10^{-3}	0.15759	54,370	8.68M	3.225 ms	18.56 MiB
Ours: Redundant+ρ	128	128	46	3.960×10^{-3}	0.11611	54,403	8.70M	3.156 ms	18.56 MiB

A.3 Reliability Aggregation Choices

We tested several simple ways to aggregate bit-level confidence over a relation support. All explicit relation statistics improve over the no-statistic control, while the log-domain variants are strongest (Table 9). MeanLog and SumLog are effectively tied, so the main method uses MeanLog for normalized scale rather than claiming unique optimality.

Table 9: **Reliability aggregation ablation on Learned$(31, 16)$.** Alternative statistics all improve over no explicit relation statistic; the log-domain variants are strongest. MeanLog is used in the main method for relation-length-normalized scale, not because it is uniquely optimal.

Statistic	3–5 dB BER	Gain vs. None
None	3.37×10^{-3}	–
MeanReliability	3.09×10^{-3}	7.86%
RawProduct	3.06×10^{-3}	9.70%
MeanLog	3.03×10^{-3}	11.13%
SumLog	3.02×10^{-3}	11.40%

A.4 Paired Rescue–Harm Details

Table 10: **Paired rescue–harm behavior of C3 over 3–5 dB.** Rescue means Redundant-None fails while Redundant+ρ succeeds; harm is the reverse. Net FER rescue is rescue rate minus harm rate; the final column is reported in percentage points.

Code	Rescue:Harm	Net FER rescue
Gallager LDPC$(96, 50)$	$8.37\times$	$+3.48$
BCH$(63, 36)$	$21.25\times$	$+3.70$
MacKay LDPC$(96, 48)$	$2.96\times$	$+0.32$

A.5 Cross-Code Same-Weight Relation–Reliability Alignment

Table 11: **Cross-code same-weight relation–reliability alignment control.** SameWeightShuffle preserves each frame's exact reliability multiset inside every equal-weight relation group and makes no cross-weight moves. BER values are means over three training seeds; alignment gain is the BER reduction of True relative to SameWeightShuffle.

Code	E_b/N_0	True	SameWeightShuffle	Gain
MacKay	3 dB	5.082×10^{-3}	1.173×10^{-2}	$56.56 \pm 2.16\%$
	4 dB	4.139×10^{-4}	2.045×10^{-3}	$79.55 \pm 2.27\%$
	5 dB	9.855×10^{-6}	2.813×10^{-4}	$96.18 \pm 1.31\%$
Gallager	3 dB	8.252×10^{-3}	3.244×10^{-2}	$74.56 \pm 0.57\%$
	4 dB	1.335×10^{-3}	1.233×10^{-2}	$89.17 \pm 0.39\%$
	5 dB	1.943×10^{-4}	3.564×10^{-3}	$94.54 \pm 0.13\%$

Across 3–5 dB, SameWeightShuffle increases the seed-level mean BER by $2.55 \pm 0.16\times$ on MacKay and $4.94 \pm 0.10\times$ on Gallager relative to the correctly aligned assignment. The earlier unrestricted MacKay WithinShuffle intervention gives the same qualitative conclusion, but uses a separately generated paired evaluation stream.

A.6 Gallager Paired Common-Stream Random-WM Re-evaluation

The paired common-stream Gallager$(96, 50)$ control performs no additional training. It loads the 500k NativeRepeat, Selected Redundant, and Selected-matched Random-WM checkpoints for training seeds 101–103 and re-evaluates all three models on exactly the same random messages and AWGN samples. For each seed and each E_b/N_0 point in 3–6 dB, the common stream contains 131,072 frames. Before evaluation, we verify that every loaded checkpoint uses the intended support matrix and that all three supports span the same rank-46 dual space.

For the exact weight-matched comparison, Selected and Random-WM both use $M = 128$, 128 unique relations, and the full row-weight histogram $\{6{:}49, 8{:}32, 10{:}47\}$, whereas NativeRepeat has mean row weight 6. Over 3–5 dB, the paired mean BERs are 5.928×10^{-3} for NativeRepeat, 4.549×10^{-3} for Random-WM, and 4.517×10^{-3} for Selected. The corresponding reductions are $23.25\% \pm 0.25\%$ for NativeRepeat→Random-WM, $23.80\% \pm 0.15\%$ for NativeRepeat→Selected, and only $0.71\% \pm 0.40\%$ for Random-WM→Selected. At 5 dB, Random-WM is slightly better than Selected, while Selected is slightly better at 3, 4, and 6 dB. We therefore treat the two matched redundant supports as essentially tied on Gallager rather than claiming selector superiority.

A.7 Gallager Same-Weight Relation-Alignment Replication

The same-weight Gallager$(96, 50)$ control performs no additional training. For seeds 101–103, it loads the existing 500k Redundant+ρ checkpoints and compares the native TRUE reliability assignment with an inference-only SAMEWEIGHTSHUFFLE intervention on exactly the same received frames. The selected support has $M = 128$, rank 46, 128 unique relations, and weight histogram $\{6{:}49, 8{:}32, 10{:}47\}$. Reliability values are permuted independently within the weight-6, weight-8, and weight-10 groups using nonzero cyclic shifts. Hence all 128 relations are shufflable, no value crosses a relation-weight class, and the exact per-frame reliability multiset within every class is preserved.

At 3, 4, and 5 dB, mean TRUE BER is 8.252×10^{-3}, 1.335×10^{-3}, and 1.943×10^{-4}, while SAMEWEIGHTSHUFFLE gives 3.244×10^{-2}, 1.233×10^{-2}, and 3.564×10^{-3}. Correct relation alignment therefore reduces BER relative to the shuffled control by 74.56%, 89.17%, and 94.54%, respectively. On the seed-level 3–5 dB mean, the shuffled condition is $4.94 \pm 0.10\times$ worse, corresponding to a $79.76\% \pm 0.43\%$ BER reduction from restoring the correct relation–reliability assignment. This reproduces the MacKay same-weight mechanism result on a second LDPC construction.

A.8 Gallager Larger-Code Equivalent-Basis Replication

The Gallager$(96, 50)$ larger-code study retrains two matched CrossMPT decoders from scratch for 500k steps. A deterministic independent subset of the historical native parity rows forms a full-rank 46×96 basis H_0. A second basis is generated as $H_1 = AH_0$ using an invertible 46×46 GF(2) transform. The dense candidate is selected from a fixed deterministic transform set using mean parity-row weight only, with no BER-based selection. Both conditions therefore have $M = 46$, rank 46, the same code, and exactly the same dual row space. Their support fingerprints are fixed across all three training seeds.

The two models use bit-identical initial parameters, identical training E_b/N_0 and channel/codeword streams, matched per-step dropout random-number-generator streams, and a common random-message/AWGN evaluation stream. The native basis has row weight 6 for all 46 rows; the dense basis has mean row weight 50.43 (range 40–62). At 3, 4, 5, and 6 dB, the seed-averaged Dense/Native BER ratios are $4.34\times$, $11.68\times$, $37.48\times$, and $96.37\times$, respectively. Over the 3–5 dB mean, NativeBasis BER is 7.41×10^{-3} and DenseEquivalent BER is 5.09×10^{-2}, giving a seed-wise degradation of $6.87 \pm 0.05\times$. Because the row-weight distributions intentionally differ, this experiment should be read as a longer-code replication of *equivalent-representation sensitivity*, not as a density-controlled attribution of the effect to basis identity alone.

A.9 Exact Row-Weight-Matched Equivalent-Basis Control

The exact weight-matched Learned$(31, 16)$ control directly removes parity-row weight distribution as a nuisance variable. Starting from the native full-rank 15×31 basis, we enumerate the nonzero dual relations and construct an alternative full-rank basis $H_1 = AH_0$ with the same $M = 15$, rank 15, dual row space, and full 31-bit coverage. Both bases have exactly the row-weight histogram $\{3{:}3, 5{:}7, 6{:}2, 7{:}3\}$ and mean row weight 5.133. The alternate basis shares 10 of 15 exact rows with the native basis; its remaining geometry is not identical (column-degree $L_1 = 18$ and pairwise-intersection-histogram $L_1 = 40$), so this experiment isolates row-weight distribution rather than all graph-geometry factors. Basis selection uses geometry only and never BER.

For seeds 42–44, the two 28,498-parameter CrossMPT models use bit-identical initialization, identical training E_b/N_0/AWGN streams, matched per-step dropout random-number-generator streams, and

identical random-message/AWGN evaluation streams. At 2, 3, 4, 5, and 6 dB, the seed-averaged alternate/native BER ratios are $1.09\times$, $1.17\times$, $1.28\times$, $1.47\times$, and $1.82\times$, respectively. Over the seed-wise 3–5 dB mean, the ratios are $1.251\times$, $1.166\times$, and $1.223\times$, giving $1.213 \pm 0.043\times$ overall, or a $21.30 \pm 4.33\%$ BER increase. Thus the large dense-basis gaps are partly amplified by graph density, but a reproducible residual sensitivity remains even when the complete parity-row weight multiset is matched exactly.

Table 12: **Exact row-weight-matched C1 control on Learned**$(31, 16)$**.** Both bases span the same rank-15 dual space and have the identical row-weight multiset. Values are means over seeds 42–44.

E_b/N_0	NativeBasis BER	WeightMatchedAlt BER	Alt/Native
2 dB	3.807×10^{-2}	4.163×10^{-2}	$1.09\times$
3 dB	1.620×10^{-2}	1.895×10^{-2}	$1.17\times$
4 dB	5.153×10^{-3}	6.617×10^{-3}	$1.28\times$
5 dB	1.163×10^{-3}	1.715×10^{-3}	$1.47\times$
6 dB	1.662×10^{-4}	3.032×10^{-4}	$1.82\times$

A.10 MacKay 300k-Horizon Global-Best Re-evaluation

The MacKay$(96, 48)$ 300k-horizon study is a *300k-horizon global-best re-evaluation*, not an exact-step-300k evaluation. NativeRepeat, Selected Redundant, and Selected Redundant+ρ are trained through 300k steps for seeds 42–44, and each condition applies the same 4 dB validation-based global-best checkpoint-selection rule. The selected checkpoints, which occur between 210k and 300k steps, are then evaluated on an independent common stream of 131,072 frames per E_b/N_0 point at 3, 4, and 5 dB. All three seeds show positive 3–5 dB C2, C3, and combined gains. A same-stream comparison against the earlier 100k checkpoints confirms that all three representation conditions continue to improve over the longer training horizon; the paper reports the measured 300k-horizon values rather than extrapolating from intermediate checkpoints.

The 3–5 dB relative gains are $9.97\% \pm 3.07\%$ for NativeRepeat$\rightarrow$Selected Redundant, $6.68\% \pm 4.90\%$ for Selected Redundant$\rightarrow$Selected Redundant+ρ, and $16.00\% \pm 4.91\%$ for the combined comparison. At one individual seed–E_b/N_0 pair (seed 43 at 4 dB), the C3 change is slightly negative, so we do not claim that C3 improves every seed–E_b/N_0 instance; the aggregate and per-seed 3–5 dB effects remain positive.

The Selected-matched Random-WM control uses the same candidate family and exact selected row-weight histogram, retains all 48 native checks, and has $M = 128$, rank 48, and 128 unique relations. Its validation-selected checkpoints occur at 280k, 300k, and 280k steps for seeds 42, 43, and 44, respectively. On the same 131,072-frame-per-E_b/N_0 evaluation family, its mean BER is 4.896×10^{-3}, 3.716×10^{-4}, and 1.168×10^{-5} at 3, 4, and 5 dB, for a 3–5 dB mean of 1.760×10^{-3}. This is a $19.49\% \pm 2.96\%$ reduction from NativeRepeat; the heuristic Selected support has $11.85\% \pm 2.31\%$ higher BER than Random-WM under the Random-WM-normalized convention.

MacKay support-draw robustness. The support-draw robustness study tests whether the matched Random-WM result could be caused by one unusually favorable support draw. It fixes the neural/training seed at 42, keeps the training, validation, and final evaluation streams unchanged, and varies only the Random-WM support across five independently seeded draws. Every draw uses the same $M = 128$, rank-48 row space, 48 native weight-6 checks, 80 non-native weight-10 relations, and excludes the 80 heuristic-Selected extras. The five 3–5 dB mean BERs lie between 1.765×10^{-3} and 1.788×10^{-3}, with mean 1.771×10^{-3} and descriptive support-draw standard deviation 9.76×10^{-6} (coefficient of variation 0.55%). All five supports outperform the fixed-seed Selected reference 2.010×10^{-3}, corresponding to an 11.90% reduction using the mean across draws. Their pairwise non-native overlap ranges from 24 to 36 of 80 relations, so the result is not five copies of nearly identical supports. We report this variability only as support-draw robustness conditional on training seed 42, not as training-seed uncertainty.

A.11 GALLAGER$(252, 128)$ SCALE REPLICATION

The scale study tests whether the C2/C3 representation ladder persists beyond the $n \leq 96$ main LDPC benchmarks. We construct a Gallager LDPC$(252, 128)$ code with 126 native weight-6 parity rows, dual rank 124, and column weight 3. All three decoder conditions use the same $M = 256$ token budget and are trained from scratch for 300k steps on seeds 101–103 with paired channel and dropout streams. *NativeRepeat* contains 126 unique native relations repeated to fill the budget. *Redundant* retains the native checks and adds 130 distinct pair-XOR dual relations, yielding 256 unique relations with weight histogram $\{6{:}127, 8{:}37, 10{:}92\}$ and mean row weight 7.73. *Redundant+ρ* uses exactly the same redundant support and adds relation-conditioned reliability. The relation set is chosen deterministically by low weight with SHA-1 tie-breaking and no BER-based tuning.

Table 13: **Gallager**$(252, 128)$ **larger-blocklength replication.** BER values are means over seeds 101–103 on 131,072 common-stream random codewords per seed and E_b/N_0 point. We define the C2 gain as the BER reduction from NativeRepeat to Redundant, the C3 gain as the additional BER reduction from Redundant to Redundant+ρ, and the total gain as the BER reduction from NativeRepeat to Redundant+ρ.

E_b/N_0	NativeRepeat	Redundant	Redundant+ρ	C2	C3	Total
3 dB	1.523×10^{-2}	1.240×10^{-2}	8.544×10^{-3}	18.56%	31.12%	43.91%
4 dB	2.856×10^{-3}	1.917×10^{-3}	9.775×10^{-4}	32.87%	49.01%	65.77%
5 dB	2.678×10^{-4}	1.382×10^{-4}	5.064×10^{-5}	48.41%	63.35%	81.09%
6 dB	1.497×10^{-5}	6.751×10^{-6}	2.018×10^{-6}	54.89%	70.10%	86.51%

Over the seed-level 3–5 dB mean, C2 reduces BER by $21.22\% \pm 0.26\%$, C3 gives a further $33.80\% \pm 0.24\%$ reduction, and the combined reduction is $47.85\% \pm 0.02\%$; all three effects are positive for all three training seeds. Because the redundant support is heavier on average than NativeRepeat (7.73 versus 6.00), the C2 scale result is not attributable to a lower average parity-row weight. We treat this as a larger-blocklength replication of the measured C2/C3 trends, not as evidence of an asymptotic scaling law from a single additional code.